CLAIMS

1. A method for optimizing design of a microelectronic circuit using a plurality of processors, the design having a plurality of timing paths, a subset of the timing paths being characterized as critical paths, each of the timing paths having an endpoint, the method comprising the steps of:

constructing a list of the critical paths;

constructing a set of endpoints of said critical paths;

partitioning said set of endpoints in accordance with predetermined rules regarding timing independence and geometric independence of the critical paths, thereby defining sub-sets of endpoints and paths associated therewith; and

optimizing timing of the critical paths,

wherein

said optimizing is performed in parallel by the processors, each of the processors optimizing timing of the paths associated with the endpoints in respective sub-sets, and

the microelectronic circuit includes a multiplicity of components, and said method is performed after placement of the components in the design.

2. A method according to claim 1, wherein

said step of constructing the set of endpoints comprises constructing an endpoint graph from the list of critical paths, the endpoint graph having at least one vertex representing critical paths associated with a given endpoint; and

said partitioning step comprises partitioning the endpoint graph to define sub-sets of vertices.

3. A method according to claim 1, wherein the rule regarding timing independence requires that all critical paths associated with a given endpoint are assigned to a same sub-set of paths, so that all critical paths associated with a given endpoint are optimized using a same processor.

4. A method according to claim 1, wherein the rule regarding geometric independence requires that all overlapping critical paths are assigned to a same sub-set of paths, so that all overlapping critical paths are optimized using a same processor.

5. A method according to claim 2, wherein the endpoint graph has at least one edge connecting two vertices, the edge representing overlap between paths associated with endpoints represented by the respective two vertices.

6. A method according to claim 5, wherein

said partitioning step is performed in a plurality of iterations;

in a given iteration, a vertex belonging to a given sub-set does not belong to any other sub-set; and

vertices belonging to distinct sub-sets are not connected by an edge.

7. A method according to claim 1, wherein said partitioning step and said optimizing step are performed in a plurality of iterations, so that critical paths not meeting said predetermined rules in a first iteration are not optimized in the first iteration.

8. A method according to claim 7, further comprising the steps of:

determining a number of iterations needed to optimize the design; and

determining a number of processors needed in each iteration, such that a maximum possible number of critical paths is equally balanced between processors.

9. A method according to claim 2, wherein

said partitioning step and said optimizing step are performed in a plurality of iterations, so that critical paths not meeting said predetermined rules in a given iteration are not optimized in that iteration, and

in a subsequent iteration of said partitioning step, said partitioning comprises constructing a remnant graph having vertices representing endpoints of non-optimized critical paths.

10. A method according to claim 2, further comprising the step, prior to constructing the endpoint graph, of constructing a graph of the critical paths characterized as a cluster graph, the cluster graph having vertices where each vertex represents a plurality of connected critical paths.

11. A method according to claim 10, wherein the cluster graph has at least one edge connecting two vertices, the edge representing overlap between areas occupied by paths represented by the respective vertices.

12. A method for optimizing design of a microelectronic circuit using a plurality of processors, the design having a plurality of timing paths, a subset of the timing paths being characterized as critical paths, each of the timing paths having an endpoint, the method comprising the steps of:

constructing a list of the critical paths;

constructing an endpoint graph from the list of critical paths, the endpoint graph having at least one vertex representing critical paths associated with a given endpoint;

partitioning the endpoint graph in accordance with predetermined rules regarding timing independence and geometric independence of the critical paths, thereby defining sub-sets of vertices of the endpoint graph; and

optimizing timing of the critical paths, said optimizing including the steps of

identifying the endpoints represented by the vertices in a given sub-set of vertices,

identifying all critical paths ending at said identified endpoints,

generating design changes in the microelectronic circuit to optimize said identified critical paths, and

storing said design changes in a memory unit,

wherein the microelectronic circuit includes a multiplicity of components, and said method is performed after placement of the components in the design.

13. A method according to claim 12, wherein said optimizing is performed in parallel by the respective processors, each processor optimizing critical paths associated with a different sub-set of vertices of the endpoint graph, so that the design changes are stored in a plurality of memory units, and further comprising the step of updating a main memory by storing therein the design changes in the respective memory units after completion of optimization by all the processors.

14. A method according to claim 13, wherein said optimizing and said updating are performed in a plurality of iterations.

15. A method according to claim 13, further comprising the step of evaluating results of said optimizing by performing a timing analysis of the critical paths.

16. A method according to claim 1, wherein each of the timing paths has an arrival time for a signal on that path, and a critical path is characterized as having an arrival time greater than a required arrival time.

17. A method according to claim 12 wherein each of the timing paths has an arrival time for a signal on that path, and a critical path is characterized as having an arrival time greater than a required arrival time.

18. A computer-readable storage medium having stored therein instructions for performing a method for optimizing design of a microelectronic circuit using a plurality of processors, the design having a plurality of timing paths, a subset of the timing paths being characterized as critical paths, each of the timing paths having an endpoint, the method comprising the steps of:

constructing a list of the critical paths;

constructing a set of endpoints of said critical paths;

partitioning said set of endpoints in accordance with predetermined rules regarding timing independence and geometric independence of the critical paths, thereby defining sub-sets of endpoints and paths associated therewith; and

optimizing timing of the critical paths,

wherein

said optimizing is performed in parallel by the processors, each of the processors optimizing timing of the paths associated with the endpoints in respective sub-sets, and

the microelectronic circuit includes a multiplicity of components, and said method is performed after placement of the components in the design.

19. A computer-readable storage medium according to claim 18, wherein

said step of constructing the set of endpoints comprises constructing an endpoint graph from the list of critical paths, the endpoint graph having at least one vertex representing critical paths associated with a given endpoint; and

said partitioning step comprises partitioning the endpoint graph to define sub-sets of vertices.

20. A computer-readable storage medium having stored therein instructions for performing a method for optimizing design of a microelectronic circuit using a plurality of processors, the design having a plurality of timing paths, a subset of the timing paths being characterized as critical paths, each of the timing paths having an endpoint, the method comprising the steps of:

constructing a list of the critical paths;

constructing an endpoint graph from the list of critical paths, the endpoint graph having at least one vertex representing critical paths associated with a given endpoint;

partitioning the endpoint graph in accordance with predetermined rules regarding timing independence and geometric independence of the critical paths, thereby defining sub-sets of vertices of the endpoint graph; and

optimizing timing of the critical paths, said optimizing including the steps of

identifying the endpoints represented by the vertices in a given sub-set of vertices,

identifying all critical paths ending at said identified endpoints,

generating design changes in the microelectronic circuit to optimize said identified critical paths, and

storing said design changes in a memory unit,

wherein the microelectronic circuit includes a multiplicity of components, and said method is performed after placement of the components in the design.

21. A computer-readable storage medium according to claim 20, wherein said optimizing is performed in parallel by the respective processors, each processor optimizing critical paths associated with a different sub-set of vertices of the endpoint graph, so that the design changes are stored in a plurality of memory units, and further comprising the step of updating a main memory by storing therein the design changes in the respective memory units after completion of optimization by all the processors.